IN THE SPECIFICATION

Please replace the following paragraph beginning on page 6, line 1, with the following rewritten paragraph:

FIG. 13 [[12]] is a block diagram of a data processing system which contains a CPU with a processor that may use an apparatus for managing processor operations according to embodiments of the present invention.

Please replace the following paragraph beginning on page 7, line 21, with the following rewritten paragraph:

FIG. 2 is a block diagram of details of IFU 101, IMU 102 and SMU 103. IFU 101 comprises an Instruction Pre-fetch Buffer 207, I-Cache 210, Millicode Read Only Memory (ROM) 211, Translation Tables 209 and Branch Tables 208. IMU 102 comprises Instruction Window Buffer (IWB) 204, Register File 205, and Translation Look-up Buffer (TLB) 223 [[203]]. SMU 103 comprises SRB 212 and TLB 203. SRB 212 is the unit that comprises embodiments of the present invention. Although all of the preceding units are used with embodiments of the present invention, the detailed operations of all the units are not necessary to explain the present invention. Only details of the necessary units will be explained in the following description.

Please replace the following paragraph beginning on page 8, line 23, with the following rewritten paragraph:

FIG. 5 illustrates details of SMU 103 according to embodiments of the present invention. IFU 101 and IMU 102 are coupled to SRB 212 in SMU 103. SMU 103 has additional units, Data Cache Unit (DCU) 501 and Data Aligner 502. D-Cache 213 is associated with DCU 501. Cache requests are sent to DCU 501 and Load packets are returned to SRB 212. Data Aligner 502 positions data bytes (operands) in the correct position for return to IMU 102 via connection 503.

Please replace the following paragraph beginning on page 10, line 12, with the following rewritten paragraph:

<u>The following details the organization of a register 600 in SRB 212.</u> Valid bit (1 bit) 601 indicates the validity of the Quadword 607 during Loads and Stores. If an operation is a Load, the Quadword 607 becomes valid as soon as the full Quadword has been fetched. For a Store, the Quadword 607 becomes valid only if the Stored operand (1, 2, 4, or 8 bytes) becomes valid and the remaining bytes have been fetched from a valid location or from the cache. Instruction ID (6 bits) 602 is used to identify the instruction associated with a Load/Store. There may be more than one entry with the same instruction ID, as when multiple Quadwords are needed to satisfy a Load or a Store. In these cases, instructions with the same instruction ID will all be adjacent. Status (3 bits) 603 indicates the status of SRB 212 entries. Load/Store (1 bit) 604 indicates whether the operation is a Load or a Store. Real Address (64 bits) 605 refers to the Quadword real address which is needed before the D-cache 213 may be accessed. Operand Mask 606 is essentially a 16-bit field that indicates selected bytes in a Quadword. Although the Operand Mask 606 is 16 bits, not all states are valid. The operand mask is on 1, 2, 4, 8 byte boundaries. Quadword operands 607 are the 16 bytes of data accessed from a D-Cache line.

Please replace the following paragraph beginning on page 11, line 4, with the following rewritten paragraph:

FIG. 6 also illustrates an exemplary 22 entries in the SRB 212. The entries are numbered for illustration, for example entry 620 is numbered 11. In FIG. 6, the IN pointer 615 is set to the entry position number 4 and the OUT pointer 614 is set to the entry position number 19 which defines a window of instructions. The entries 5 through 18 are the entries that are currently valid indicated by a one in the Valid bit field 601. Since the instruction IDs (IID) <u>602 are</u> [[602are]] known and supplied when the instructions are dispatched from the IFU 101 to the SMU 103, the IID field 602 of all the entries in the "window"(defined by IN pointer 614 and OUT pointer 615) are shown with IID (e.g., IID 619). Table 1 (on pg. 21) illustrates conditions for setting status bits in the

Instruction Status field 603. All the instructions in the SRB 212 must be either a Load or a Store indicated by a "Load " (e.g., 617) or a "Store" (e.g., 616) in the IID field 602 of FIG. 6. In actual practice, a logic one or zero would be used for this designation. Real addresses are generated only after the IMU 102 calculates the virtual address and the TLB 203 generates the appropriate real address (e.g., 618). Since address calculation is in a decoupled unit, the order in which real addresses are filled into the Real Address field 605 is not predictable. However, once the address does get filled in an entry, that entry is said to have its address "resolved". The Operand Mask 606 is used to determine which bytes of the Quadword operand field 607 are selected for a particular Load or Store operation. Use of the Operand Mask 606 is detailed later in the disclosure. The Quadword operands 608 through 613 are used to describe states of entries within the window defined by IN pointer 615 and OUT pointer 614. The Quadword field 607 is shown for the specific Quadwords as numbered blocks 0 through 16, where each block represents a byte of data.

Please replace the following paragraph beginning on page 12, line 3, with the following rewritten paragraph:

Typically an SRB 212 entry starts without any operand and this case is indicated by blank Quadword operand entries (e.g., Quadword operand 621). For a Store operation, the base part of the Quadword operand must be acquired. The base part of the Quadword operand field 607 is the part that is unaffected by the Store operation. That part of the Quadword operand field 607 that is to be changed is called just the "operand". The update of these two parts of the Quadword operand field 607 occurs [[~~occurS~~]] independently and their order is not predictable. Quadword Operand 608 and 609 illustrate these two cases. In Quadword Operand 608, the base (bytes 0-3, and 8-15) is updated first. The "operand" part (bytes 4-7) is updated second. In Quadword Operand 609, the "operand" (bytes 8-11) is updated first and the base (bytes 0-7 and 12-15) is updated second. Eventually both parts of the Quadword Operand field 607 are updated resulting in a completed Quadword Operand (e.g., Quadword Operand 611) which is shown completely hatched. An entire Quadword Operand field 607 of Quadword

Operands [[OperandS]] 610 and 612 are shown shaded to indicate an entire update of the Quadword Operand for a Load operation. After the Quadword Operand field 607 is updated, the Load needs to be "issued" to transfer the appropriate part of the Quadword Operand field 607 to the IMU 102. Only after an issue takes place can the Load entry be marked as completed. The Load in entry position 18 has Quadword Operand 613 which is double cross-hatched to show an entire Quadword that has been updated but as yet has not be "issued".

Please replace the following paragraph beginning on page 12, line 23, with the following rewritten paragraph:

FIG. 9 is a flow diagram of steps used in embodiments of the present to perform a Store operation using SRB 212. In step 901, the IWB 204 issues an address generation (AGEN) instruction to ATU 403. The real address is obtained in step 902 and in step 903 the address field is updated. This address updating step updates real address 914. In step 904, the SRB 212 issues a Quadword fetch both to itself and to the D-Cache 213. It is possible that the desired Quadword is already in the SRB 212 from a previous Load. In step 905, the Quadword is received. The Operand Mask Data 915 is derived from lower order real address bits. The operand mask indicates which bytes in the Quadword are to be Stored. The D-Cache 213 has those Quadword operands that are not to be changed and the addresses/operands 105 from the IMU 102 supplies the operands that are to be changed by the Store operation. Since it is not known whether the entry to the SRB 212 or the retrieval of the Load packet from the D-Cache 213 will occur first, overwriting the SRB 212 entry with the D-Cache 213 Quadword must be protected. Therefore, in step 906 the Quadword under a complement operand mask (e.g., 911 and 913) is updated with the Quadword from D-Cache. In step 907, the IWB 204 sends the Store operand. In step 908, the Store operand is rotated into position in the Quadword, and in step 909 the Quadword is updated under the operand mask (e.g., 916) [[(e.g., 911)]] with the Store operand. In step 910, the entire Quadword 912 is updated with the Store operands and the Store is completed.

Please replace the following paragraph beginning on page 13, line 19, with the following rewritten paragraph:

FIG. 10 is a flow diagram of a Load operation. In step 1001, the IWB 204 issues a AGEN instruction to the ATU 403 and the real address is generated in step 1002. In step 1003, the real address field 1009 is updated in the SRB 212. In step 1004, the SRB 212 issues a Quadword fetch to itself and D-Cache 213. In step 1005, the Quadword 1010 is received from either the SRB 212 or the D-Cache 213. In step 1006 the SRB 212 or the D-Cache 213 sends the Quadword <u>1010</u>. In step 1007, the operand <u>1012</u> is extracted using the Operand Mask <u>Data 1011</u> [[1010]] and rotated and sign extended if necessary. In step 1008, the IWB 204 receives the loaded operand <u>1012</u>.

Please replace the following paragraph beginning on page 16, line 11, with the following rewritten paragraph:

FIG. 12 is a state transition diagram further explaining a Store operation according to embodiments of the present invention. The Store operation is slightly different from the Load operation because two actions have to be completed, namely those of updating the base part (that which remains unchanged) of the Quadword field and the operand or data part (that which changes) of a Quadword 912. <u>EE 1205 indicates that the Store operation has established entry to SRB 212.</u> The base part is updated from the cache (e.g., D-Cache 213) or from another SRB 212 entry as with a Load operation. The data part is updated in the transition DA 1215. AA 1206 performs steps 901, 902, and 903 of FIG. 9. As in a Load operation, the RC 1207 filter for a Store operation performs steps 904, 905, and 906 of FIG. 9. The QA 1208 filter performs steps 904, 905, and 906 of FIG. 9. Transition QA 1208 occurs when a Quadword arrives. The SC 1214 transition occurs when the Quadword is sent to D-cache 213 and the entry is purged. Transition SC 1214 transitions to the Store Completed [quadword]<u>Quadword</u> valid state 1213. The transition DA 1215 (data arrives) causes a transition from the Unresolved data invalid 1201 to Unresolved data valid 1209, Address resolved data invalid 1202 to Address resolved data valid 1210, Request Issued data invalid 1203 to Request Issued data valid 1211 and Base valid data invalid 1204 to Base valid data valid 1212.

Please replace the following paragraph beginning on page 20, line 6, with the following rewritten paragraph:

FIG. 8 is a flow diagram of filling the real address field (e.g., 605). When the IWB 204 issues an AGEN request in step 801, the address is computed and sent through TLB 203 along with the Instruction ID (IID) 602 of the request and the length of the request (number of bytes) in step 803. If translation is successful, the real address of the enclosing Quadword, a mask and the IID are sent to the SRB 212 in step 804. Since this is a common occurrence it is shown as an common path 807. The matching entry in the active window of the SRB 212 is updated to reflect the real address for the entry and the status of the entry is changed to resolved in step 805. The mask is used to set the Operand Mask 606 in the entry.

Please replace the following paragraph beginning on page 32, line 24, with the following rewritten paragraph:

Referring to FIG. 13, an example is shown of a data processing system 1300 which may be used for the invention. The system has a central processing unit (CPU) 1310, which is coupled to various other components by system bus 1312. Read-only memory ("ROM") 1316 is coupled to the system bus 1312 and includes a basic input/output system ("BIOS") that controls certain basic functions of the data processing system 1300. Random access memory ("RAM") 1314, I/O adapter 1318, and communications adapter 1334 are also coupled to the system bus 1312. I/O adapter 1318 may be a small computer system interface ("SCSI") adapter that communicates with a disk storage device 1320. Communications adapter 1334 interconnects bus 1312 with an outside network enabling the data processing system to communicate with other such systems. Input/Output devices are also connected to system bus 1312 via user interface adapter 1322. Keyboard 1324, track ball 1332 [[1323]], mouse 1326 [[1323]] and speaker 1328 are all interconnected to bus 1312 via user interface adapter 1322. In this manner, a user is capable of inputting to the system through the keyboard 1324, trackball 1332 or mouse 1326 [[1323]] and receiving output from the system via speaker 1328 and display 1338. CPU 1310 in data processing system 1300 may employ

a processor to manage Load and Stores using a SRB 212 operating according to embodiments of the present invention.